(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 113014618 B
(45)授权公告日 2022.04.29

(21)申请号 202110045400.4
(22)申请日 2021.01.12
(65)同一申请的已公布的文献号
申请公布号 CN 113014618 A
(43)申请公布日 2021.06.22
(73)专利权人 腾讯科技(深圳)有限公司
地址 518057 广东省深圳市南山区高新区科技中一路腾讯大厦35层
(72)发明人 李志涛
(74)专利代理机构 深圳市隆天联鼎知识产权代理有限公司 44232
代理人 叶虹
(51)Int.Cl.
H04L 67/566(2022.01)
H04L 49/90(2022.01)
H04L 67/1074(2022.01)
H04L 67/1097(2022.01)

(56)对比文件
CN 111190747 A,2020.05.22
CN 109905431 A,2019.06.18
CN 109271417 A,2019.01.25
CN 111381987 A,2020.07.07
CN 113014618 A,2021.06.22
US 2015199415 A1,2015.07.16

审查员 杨盈霄

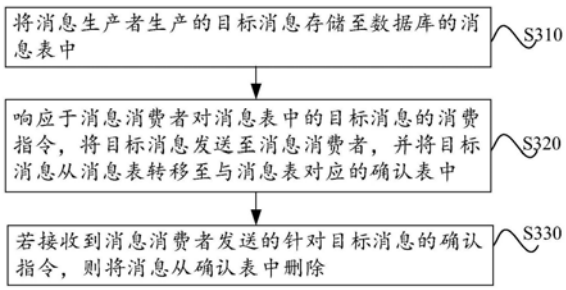
权利要求书3页 说明书14页 附图4页

(54)发明名称

消息处理方法、系统和电子设备

(57)摘要

本申请提供了一种消息处理方法、系统和电子设备，应用于计算机技术领域。该方法包括：将消息生产者生产的目标消息存储至数据库的消息表中；响应于消息消费者对消息表中的目标消息的消费指令，将目标消息发送至消息消费者，并将目标消息从消息表转移至与消息表对应的确认表中；若接收到消息消费者发送的针对目标消息的确认指令，则将消息从确认表中删除，实现使用数据库存储消息生成消息队列，能够实现使用数据库来提供消息队列功能，能够降低用户运维和使用消息队列的难度，简单易用。



CN 113014618 B

1.一种消息处理方法,其特征在于,包括:

将消息生产者生产的目标消息存储至数据库的消息表中;

响应于消息消费者对所述消息表中的所述目标消息的消费指令,将所述目标消息发送至所述消息消费者,并将所述目标消息从所述消息表转移至与所述消息表对应的确认表中;

统计所述目标消息在所述确认表中的时长;

若所述时长达到超时阈值,则统计所述目标消息被发送至所述消息消费者的发送次数;

若所述发送次数达到发送次数阈值,则将所述目标消息由所述确认表转存至过期消息存储表中,对所述过期消息存储表中的目标消息进行修复,修复后转存至所述消息表中,以再次响应所述目标消息的消费指令后重新发送所述目标消息;

若接收到所述消息消费者返回的针对所述目标消息的确认指令,则将所述消息从所述确认表中删除。

2.根据权利要求1所述的消息处理方法,其特征在于,所述将消息生产者生产的目标消息存储至数据库的消息表中,包括:

获取所述目标消息的消息特征;

基于所述消息特征,将所述目标消息存储至与所述消息特征对应的消息表。

3.根据权利要求2所述的消息处理方法,其特征在于,所述数据库中包含有多个消息队列,每个所述消息队列包含相互对应的一个消息表和一个确认表,所述基于所述消息特征,将所述目标消息存储至与所述消息特征对应的消息表,包括:

从所述数据库的消息队列元信息表中,获取所述多个消息队列的特征;

基于所述消息特征和所述多个消息队列的特征,在所述多个消息队列中选取与所述目标消息的消息特征对应的消息队列;

将所述目标消息存储至与所述目标消息的消息特征对应的消息队列包含的消息表中。

4.根据权利要求3所述的消息处理方法,其特征在于,在若接收到所述消息消费者返回的针对所述目标消息的确认指令,则将所述消息从所述确认表中删除之前,所述方法还包括:监听所述消息表的消息表特征,若所述消息表特征发生变化,且变化后的消息表特征与所述消息表所属的消息队列的特征不对应,则对所述消息消费者进行告警;

监听所述确认表的确认表特征,若所述确认表特征发生变化,且变化后的确认表特征与所述确认表所属的消息队列的特征不对应,则对所述消息消费者进行告警。

5.根据权利要求1所述的消息处理方法,其特征在于,所述方法还包括:

若所述发送次数未达到所述发送次数阈值,则将所述目标消息由所述确认表转存至与所述确认表对应的消息表中。

6.根据权利要求1所述的消息处理方法,其特征在于,所述将消息生产者生产的目标消息存储至数据库的消息表中,包括:

将所述目标消息转换成字节格式;

将字节格式的目标消息存储至所述消息表中。

7.根据权利要求1所述的消息处理方法,其特征在于,在将所述目标消息发送至所述消息消费者之后,所述方法还包括:

在所述消息消费者消费所述目标消息的过程中，禁止将所述目标消息发送至其他消费者。

8.一种消息处理系统，其特征在于，包括：

消息队列库，存储有消息生产者和消息消费者，为所述消息生产者提供生产接口，为所述消息消费者提供消费接口和确认接口；

数据库，与所述消息队列库进行消息交互，存储有相互对应的消息表、确认表和过期消息存储表，所述消息表用于存储所述消息生产者通过所述生产接口发送至所述数据库且未被发送至所述消息消费者的消息，所述确认表用于存储所述数据库通过所述消费接口发送至所述消息消费者且未被所述消息消费者确认的消息，所述过期消息存储表用于存储所述确认表转存的目标消息，其中，当统计所述目标消息在所述确认表中的时长，若所述时长达到超时阈值，则统计所述目标消息被发送至所述消息消费者的发送次数，若所述发送次数达到发送次数阈值，则将所述目标消息由所述确认表转存至过期消息存储表中，所述消息表还用于所述过期消息存储表转存的目标消息，其中，当对所述过期消息存储表中的目标消息进行修复，修复后转存至所述消息表中，以再次响应所述目标消息的消费指令后重新发送所述目标消息；所述确认表接收所述消息消费者通过所述确认接口发送的确认指令后将所述确认指令对应的消息删除；

队列管理者，用于监控所述数据库及管理所述数据库。

9.一种消息处理装置，其特征在于，包括：

存储模块，配置为将消息生产者生产的目标消息存储至数据库的消息表中；

转移模块，配置为响应于消息消费者对所述消息表中的所述目标消息的消费指令，将所述目标消息发送至所述消息消费者，并将所述目标消息从所述消息表转移至与所述消息表对应的确认表中；统计所述目标消息在所述确认表中的时长；若所述时长达到超时阈值，则统计所述目标消息被发送至所述消息消费者的发送次数；若所述发送次数达到发送次数阈值，则将所述目标消息由所述确认表转存至过期消息存储表中，对所述过期消息存储表中的目标消息进行修复，修复后转存至所述消息表中，以再次响应所述目标消息的消费指令后重新发送所述目标消息；

删除模块，配置为若接收到所述消息消费者发送的针对所述目标消息的确认指令，则将所述消息从所述确认表中删除。

10.根据权利要求9所述的装置，其特征在于，所述存储模块配置为：获取所述目标消息的消息特征；基于所述消息特征，将所述目标消息存储至与所述消息特征对应的消息表。

11.根据权利要求10所述的装置，其特征在于，所述数据库中包含有多个消息队列，每个所述消息队列包含相互对应的一个消息表和一个确认表，所述存储模块配置为：从所述数据库的消息队列元信息表中，获取所述多个消息队列的特征；基于所述消息特征和所述多个消息队列的特征，在所述多个消息队列中选取与所述目标消息的消息特征对应的消息队列；将所述目标消息存储至与所述目标消息的消息特征对应的消息队列包含的消息表中。

12.根据权利要求11所述的装置，其特征在于，所述转移模块还配置为：在若接收到所述消息消费者返回的针对所述目标消息的确认指令，则将所述消息从所述确认表中删除之前，监听所述消息表的消息表特征，若所述消息表特征发生变化且变化后的消息表特征与

所述消息表所属的消息队列的特征不对应，则对所述消息消费者进行告警；监听所述确认表的确认表特征，若所述确认表特征发生变化且变化后的确认表特征与所述确认表所属的消息队列的特征不对应，则对所述消息消费者进行告警。

13.根据权利要求10所述的装置，其特征在于，所述转移模块还配置为：若所述发送次数未达到所述发送次数阈值，则将所述目标消息由所述确认表转存至与所述确认表对应的消息表中。

14.根据权利要求10所述的装置，其特征在于，所述存储模块配置为：将所述目标消息转换成字节格式；将字节格式的目标消息存储至所述消息表中。

15.根据权利要求10所述的装置，其特征在于，所述转移模块配置为：在所述消息消费者消费所述目标消息的过程中，禁止将所述目标消息发送至其他消费者。

16.一种电子设备，其特征在于，包括：

处理器；

存储器，所述存储器上存储有计算机可读指令，所述计算机可读指令被所述处理器执行时，实现如上述权利要求1-7中任一项所述的方法。

17.一种计算机可读程序介质，其存储有计算机程序指令，当所述计算机程序指令被计算机执行时，使计算机执行权利要求1-7中任一项所述的方法。

消息处理方法、系统和电子设备

技术领域

[0001] 本申请涉及计算机及通信技术领域，特别涉及一种消息处理方法、系统和电子设备。

背景技术

[0002] 消息队列(Message Queue)是消息的传输过程中保存消息的容器，在各个业务的后台系统应用非常广泛。

[0003] 业界常用的消息队列系统有Kafka、ActiveMQ、RabbitMQ等，这些消息队列系统都使用多台服务器缓存消息，部署和维护这些系统需要较高的人力和物力成本，而且这些系统使用起来也相当复杂，用户需要了解其设计和实现，增加了使用成本。故目前消息队列系统的使用和运维相当复杂。

发明内容

[0004] 本申请旨在提供一种消息处理方法、系统和电子设备，其能够实现使用数据库来提供消息队列功能，能够降低用户运维和使用消息队列的难度，简单易用。

[0005] 根据本申请实施例的一个方面，提供了一种消息处理方法，包括：将消息生产者生产的目标消息存储至数据库的消息表中；响应于消息消费者对所述消息表中的所述目标消息的消费指令，将所述目标消息发送至所述消息消费者，并将所述目标消息从所述消息表转移至与所述消息表对应的确认表中；若接收到所述消息消费者发送的针对所述目标消息的确认指令，则将所述消息从所述确认表中删除。

[0006] 根据本申请实施例的一个方面，提供了一种消息处理装置，包括：存储模块，配置为将消息生产者生产的目标消息存储至数据库的消息表中；转移模块，配置为响应于消息消费者对所述消息表中的所述目标消息的消费指令，将所述目标消息发送至所述消息消费者，并将所述目标消息从所述消息表转移至与所述消息表对应的确认表中；删除模块，配置为若接收到所述消息消费者发送的针对所述目标消息的确认指令，则将所述消息从所述确认表中删除。

[0007] 在本申请的一个实施例中，基于前述方案，所述存储模块配置为：获取所述目标消息的消息特征；基于所述消息特征，将所述目标消息存储至与所述消息特征对应的消息表。

[0008] 在本申请的一个实施例中，基于前述方案，所述数据库中包含有多个消息队列，每个所述消息队列包含相互对应的一个消息表和一个确认表，所述存储模块配置为：从所述数据库的消息队列元信息表中，获取所述多个消息队列的特征；基于所述消息特征和所述多个消息队列的特征，在所述多个消息队列中选取与所述目标消息的消息特征对应的消息队列；将所述目标消息存储至与所述目标消息的消息特征对应的消息队列包含的消息表中。

[0009] 在本申请的一个实施例中，基于前述方案，所述转移模块还配置为：在若接收到所述消息消费者返回的针对所述目标消息的确认指令，则将所述消息从所述确认表中删除之

前，监听所述消息表的消息表特征，若所述消息表特征发生变化且变化后的消息表特征与所述消息表所属的消息队列的特征不对应，则对所述消息消费者进行告警；监听所述确认表的确认表特征，若所述确认表特征发生变化且变化后的确认表特征与所述确认表所属的消息队列的特征不对应，则对所述消息消费者进行告警。

[0010] 在本申请的一个实施例中，基于前述方案，所述转移模块还配置为：在若接收到所述消息消费者返回的针对所述目标消息的确认指令，则将所述消息从所述确认表中删除之前，统计所述目标消息在所述确认表中的时长；若所述时长达到超时阈值，则统计所述目标消息被发送至所述消息消费者的发送次数；若所述发送次数达到发送次数阈值，则将所述目标消息由所述确认表转存至过期消息存储表中。

[0011] 在本申请的一个实施例中，基于前述方案，所述转移模块还配置为：若所述发送次数未达到所述发送次数阈值，则将所述目标消息由所述确认表转存至与所述确认表对应的消息表中。

[0012] 在本申请的一个实施例中，基于前述方案，所述存储模块配置为：将所述目标消息转换成字节格式；将字节格式的目标消息存储至所述消息表中。

[0013] 在本申请的一个实施例中，基于前述方案，所述转移模块配置为：在所述消息消费者消费所述目标消息的过程中，禁止将所述目标消息发送至其他消费者。

[0014] 根据本申请实施例的一个方面，提供了一种消息处理系统，包括：消息队列库，存储有消息生产者和消息消费者，为所述消息生产者提供生产接口，为所述消息消费者提供消费接口和确认接口；数据库，与所述消息队列库进行消息交互，存储有相互对应的消息表和确认表，所述消息表用于存储所述消息生产者通过所述生产接口发送至所述数据库且未被发送至所述消息消费者的消息，所述确认表用于存储所述数据库通过所述消费接口发送至所述消息消费者且未被所述消息消费者确认的消息，所述确认表接收所述消息消费者通过所述确认接口发送的确认指令后将所述确认指令对应的消息删除；队列管理者，用于监控所述数据库及管理所述数据库。

[0015] 根据本申请实施例的一个方面，提供了一种计算机可读程序介质，其存储有计算机程序指令，当所述计算机程序指令被计算机执行时，使计算机执行上任一项所述的方法。

[0016] 根据本申请实施例的一个方面，提供了一种电子设备，包括：处理器；存储器，所述存储器上存储有计算机可读指令，所述计算机可读指令被所述处理器执行时，实现如上任一项所述的方法。

[0017] 根据本申请实施例的一个方面，提供了一种计算机程序产品或计算机程序，该计算机程序产品或计算机程序包括计算机指令，该计算机指令存储在计算机可读存储介质中。计算机设备的处理器从计算机可读存储介质读取该计算机指令，处理器执行该计算机指令，使得该计算机设备执行上述各种可选实施例中提供的消息处理方法。

[0018] 本申请的实施例提供的技术方案可以包括以下有益效果：

[0019] 在本申请的一些实施例所提供的技术方案中，通过将消息生产者生产的目标消息存储至数据库的消息表中，响应于消息消费者对消息表中的目标消息的消费指令，将目标消息发送至消息消费者，并将目标消息从消息表转移至与消息表对应的确认表中，若接收到消息消费者发送的针对目标消息的确认指令，则将消息从确认表中删除，实现使用数据库存储消息生成消息队列，并且还通过数据库实现了向消息队列中生产消息和从消息队列

中消费消息的功能，相比于使用多台服务器存储消息，能够实现使用数据库来提供消息队列功能，能够降低用户运维和使用消息队列的难度，简单易用。

[0020] 应当理解的是，以上的一般描述和后文的细节描述仅是示例性的，并不能限制本申请。

附图说明

[0021] 此处的附图被并入说明书中并构成本说明书的一部分，示出了符合本申请的实施例，并于说明书一起用于解释本申请的原理。

[0022] 图1A示出了可以应用本申请一个实施例的技术方案的数据共享系统的示意图；

[0023] 图1B示出了可以应用本申请一个实施例的区块链示意图；

[0024] 图1C示出了可以应用本申请一个实施例的区块链中新区块生成的示意图；

[0025] 图2示意性示出了可以应用本申请实施例的技术方案的示例性系统架构的示意图；

[0026] 图3示意性示出了根据本申请的一个实施例的消息处理方法的流程图；

[0027] 图4示意性示出了根据本申请的一个实施例的消息处理系统的结构示意图；

[0028] 图5示意性示出了根据本申请的一个实施例的消息处理装置的框图；

[0029] 图6是根据一示例性实施例示出的一种电子设备的硬件框图。

具体实施方式

[0030] 现在将参考附图更全面地描述示例实施方式。然而，示例实施方式能够以多种形式实施，且不应被理解为限于在此阐述的范例；相反，提供这些实施方式使得本申请将更加全面和完整，并将示例实施方式的构思全面地传达给本领域的技术人员。

[0031] 此外，所描述的特征、结构或特性可以以任何合适的方式结合在一个或更多实施例中。在下面的描述中，提供许多具体细节从而给出对本申请的实施例的充分理解。然而，本领域技术人员将意识到，可以实践本申请的技术方案而没有特定细节中的一个或更多，或者可以采用其它的方法、组元、装置、步骤等。在其它情况下，不详细示出或描述公知方法、装置、实现或者操作以避免模糊本申请的各方面。

[0032] 附图中所示的方框图仅仅是功能实体，不一定必须与物理上独立的实体相对应。即，可以采用软件形式来实现这些功能实体，或在一个或多个硬件模块或集成电路中实现这些功能实体，或在不同网络和/或处理器装置和/或微控制器装置中实现这些功能实体。

[0033] 附图中所示的流程图仅是示例性说明，不是必须包括所有的内容和操作/步骤，也不是必须按所描述的顺序执行。例如，有的操作/步骤还可以分解，而有的操作/步骤可以合并或部分合并，因此实际执行的顺序有可能根据实际情况改变。

[0034] 云技术(Cloud technology)是指在广域网或局域网内将硬件、软件、网络等系列资源统一起来，实现数据的计算、储存、处理和共享的一种托管技术。

[0035] 云技术(Cloud technology)基于云计算商业模式应用的网络技术、信息技术、整合技术、管理平台技术、应用技术等的总称，可以组成资源池，按需所用，灵活便利。云计算技术将变成重要支撑。技术网络系统的后台服务需要大量的计算、存储资源，如视频网站、图片类网站和更多的门户网站。伴随着互联网行业的高度发展和应用，将来每个物品都有

可能存在自己的识别标志,都需要传输到后台系统进行逻辑处理,不同程度级别的数据将会分开处理,各类行业数据皆需要强大的系统后盾支撑,只能通过云计算来实现。

[0036] 数据库(Database),简而言之可视为电子化的文件柜——存储电子文件的处所,用户可以对文件中的数据进行新增、查询、更新、删除等操作。所谓“数据库”是以一定方式储存在一起、能与多个用户共享、具有尽可能小的冗余度、与应用程序彼此独立的数据集合。

[0037] 数据库管理系统(英语:Database Management System,简称DBMS)是为管理数据库而设计的电脑软件系统,一般具有存储、截取、安全保障、备份等基础功能。数据库管理系统可以依据它所支持的数据库模型来作分类,例如关系式、XML(Extensible Markup Language,即可扩展标记语言);或依据所支持的计算机类型来作分类,例如服务器群集、移动电话;或依据所用查询语言来作分类,例如SQL(结构化查询语言(Structured Query Language)、XQuery;或依据性能冲量重点来作分类,例如最大规模、最高运行速度;亦或其他的分类方式。不论使用哪种分类方式,一些DBMS能够跨类别,例如,同时支持多种查询语言。

[0038] 数据库可以应用于自动化部署消息队列,由于其具有轻量级、易维护的特点,能够显著降低生成消息队列的复杂程度。

[0039] 在本申请的一个实施例中,数据库可以存储在图1A所示的数据共享系统中,参见图1A,数据共享系统100是指用于进行节点与节点之间数据共享的系统,该数据共享系统100中可以包括多个节点101,多个节点101可以是指数据共享系统100中各个客户端。每个节点101在进行正常工作可以接收到输入信息,并基于接收到的输入信息维护该数据共享系统内的共享数据。为了保证数据共享系统内的信息互通,数据共享系统中的每个节点之间可以存在信息连接,节点之间可以通过上述信息连接进行信息传输。例如,当数据共享系统中的任意节点接收到输入信息时,数据共享系统中的其他节点便根据共识算法获取该输入信息,将该输入信息作为共享数据中的数据进行存储,使得数据共享系统中全部节点上存储的数据均一致。

[0040] 对于数据共享系统中的每个节点,均具有与其对应的节点标识,而且数据共享系统中的每个节点均可以存储有数据共享系统中其他节点的节点标识,以便后续根据其他节点的节点标识,将生成的区块广播至数据共享系统中的其他节点。每个节点中可维护一个如下表所示的节点标识列表,将节点名称和节点标识对应存储至该节点标识列表中。其中,节点标识可为IP(Internet Protocol,网络之间互联的协议)地址以及其他任一种能够用于标识该节点的信息,表1中仅以IP地址为例进行说明。

[0041]

节点名称	节点标识
节点1	117.114.151.174
节点2	117.116.189.145
…	…
节点N	119.123.789.258

[0042] 数据共享系统中的每个节点均存储一条相同的区块链。区块链由多个区块组成,参见图1B,区块链由多个区块组成,创始块中包括区块头和区块主体,区块头中存储有输入信息特征值、版本号、时间戳和难度值,区块主体中存储有输入信息;创始块的下一区块以

创始块为父区块，下一区块中同样包括区块头和区块主体，区块头中存储有当前区块的输入信息特征值、父区块的区块头特征值、版本号、时间戳和难度值，并以此类推，使得区块链中每个区块中存储的区块数据均与父区块中存储的区块数据存在关联，保证了区块中输入信息的安全性。

[0043] 在生成区块链中的各个区块时，参见图1C，区块链所在的节点在接收到输入信息时，对输入信息进行校验，完成校验后，将输入信息存储至内存池中，并更新其用于记录输入信息的哈希树；之后，将更新时间戳更新为接收到输入信息的时间，并尝试不同的随机数，多次进行特征值计算，使得计算得到的特征值可以满足下述公式：

[0044] SHA256(SHA256(version+prev_hash+merkle_root+ntime+nbits+x))＜TARGET

[0045] 其中，SHA256为计算特征值所用的特征值算法；version(版本号)为区块链中相关区块协议的版本信息；prev_hash为当前区块的父区块的区块头特征值；merkle_root为输入信息的特征值；ntime为更新时间戳的更新时间；nbits为当前难度，在一段时间内为定值，并在超出固定时间段后再次进行确定；x为随机数；TARGET为特征值阈值，该特征值阈值可以根据nbits确定得到。

[0046] 这样，当计算得到满足上述公式的随机数时，便可将信息对应存储，生成区块头和区块主体，得到当前区块。随后，区块链所在节点根据数据共享系统中其他节点的节点标识，将新生成的区块分别发送给其所在的数据共享系统中的其他节点，由其他节点对新生成的区块进行校验，并在完成校验后将新生成的区块添加至其存储的区块链中，以保证数据库中存储消息的准确性。

[0047] 本申请实施例中所涉及到的消息处理方法、系统和电子设备可以基于数据库来降低生成消息队列的复杂程度，详细说明如下：

[0048] 图2示出了可以应用本申请实施例的技术方案的示例性系统架构200的示意图。

[0049] 如图2所示，系统架构200可以包括消息生产者客户端201、消息消费者客户端202、网络203和服务器204。网络203用以在消息生产者客户端201、消息消费者客户端202和服务器204之间提供通信链路的介质。网络203可以包括各种连接类型，例如有线通信链路、无线通信链路等等。

[0050] 应该理解，图2中的消息生产者客户端201、消息消费者客户端202网络203和服务器204的数目仅仅是示意性的。根据实现需要，可以具有任意数目的消息生产者客户端201、消息消费者客户端202网络203和服务器204。比如服务器204服务器可以是独立的物理服务器，也可以是多个物理服务器构成的服务器集群或者分布式系统，还可以是提供云服务、云数据库、云计算、云函数、云存储、网络服务、云通信、中间件服务、域名服务、安全服务、CDN(Content Delivery Network，内容分发网络)、以及大数据和人工智能平台等基础云计算服务的云服务器。

[0051] 其中，上述客户端(消息生产者客户端201、消息消费者客户端202)可以是终端，如智能手机、平板电脑、笔记本电脑、台式计算机、智能音箱、智能手表等，但并不局限于此。客户端以及服务器可以通过有线或无线通信方式进行直接或间接地连接，本申请在此不做限制。

[0052] 在本申请的一个实施例中，服务器204可以获取消息生产者通过消息生产者客户端201生产的目标消息，将消息生产者生产的目标消息存储至数据库的消息表中，响应于消

息消费者通过消息消费者客户端202对消息表中的目标消息的消费指令，将目标消息发送至消息消费者，并将目标消息从消息表转移至与消息表对应的确认表中，若接收到消息消费者发送的针对目标消息的确认指令，则将消息从确认表中删除，实现使用数据库存储消息生成消息队列，并且还通过数据库实现了向消息队列中生产消息和从消息队列中消费消息的功能，相比于使用多台服务器存储消息，能够实现使用数据库来提供消息队列功能，能够降低用户运维和使用消息队列的难度，简单易用。需要说明的是，本申请实施例所提供的消息处理方法一般由服务器204执行，相应地，消息处理装置一般设置于服务器204中。但是，在本申请的其它实施例中，消息生产者客户端201和消息消费者客户端202也可以与服务器204具有相似的功能，从而执行本申请实施例所提供的消息处理方法。

[0053] 以下对本申请实施例的技术方案的实现细节进行详细阐述：

[0054] 图3示意性示出了根据本申请的一个实施例的消息处理方法的流程图，该消息处理方法的执行主体可以是服务器，比如可以是图2中所示的服务器204。

[0055] 参照图3所示，该消息处理方法至少包括步骤S310至步骤S330，详细介绍如下：

[0056] 在步骤S310中，将消息生产者生产的目标消息存储至数据库的消息表中。

[0057] 在本申请的一个实施例中，消息生产者可以是生产消息的客户端中的应用程序，比如可以是图2中的客户端201中的应用程序，消息生产者可以从客户端201中获取目标消息存储至数据库的消息表中，或者消息生产者可以获取客户端201的用户操作客户端201的行为以及客户端201从互联网上下载的其他用户的行为，基于用户行为生成目标消息，将目标消息存储至数据库中，其中，消息生产者将目标消息存储在数据库中的过程被称为消息的生产过程。数据库可以是存储在服务器204中的数据库。消息消费者可以是获取消息的客户端中的应用程序，比如可以是图2中的客户端202中的应用程序，消息消费者可以通过客户端202对服务器204发送消费指令，消费指令中携带目标消息的标识，响应于消费指令向客户端202发送目标消息，以实现目标消息从客户端201传递至客户端202，其中，消息消费者从数据库中获取目标消息的过程称为消费目标消息的过程。在本申请的另一个实施例中，消息生产者也可以是上述生产消息的客户端，消息消费者也可以是上述获取消息的客户端。

[0058] 在本申请的一个实施例中，目标消息可以是基于用户的行为数据产生的，可以是消息生产者从电商系统获得的用户“收藏”、“下单”、“付款”等行为后，生成目标消息。

[0059] 在本申请的一个实施例中，目标消息可以是基于消息生产者采集或接收的其他数据产生的，在此不做限制。

[0060] 在本申请的一个实施例中，可以获取目标消息的消息特征，基于消息特征，将目标消息存储至与消息特征对应的消息表(message_table)。

[0061] 在本申请的一个实施例中，消息特征可以包括消息标识、消息类型、消息长度、消息被消费的次数、消息在消息表中的排序、消息在确认表中的排序以及消息的生产时间等。

[0062] 在本申请的一个实施例中，数据库中可以包含有多个消息队列，每个消息队列可以包含一个消息表，可以从数据库的消息队列元信息表(queue_meta_table)中，获取多个消息队列的特征，基于消息特征和多个消息队列的特征，在多个消息队列中选取与目标消息的消息特征对应的消息队列，将目标消息存储至与目标消息的消息特征对应的消息队列包含的消息表中。

[0063] 在本申请的一个实施例中，消息队列的特征可以包括消息队列中存储消息的标识包含的特定字符、消息队列中存储消息类型、消息队列中存储消息长度、消息队列中存储消息的时长、消息队列中消息最多被转存次数、用于表示消息队列中消息最多被发送至消息消费者次数的发送次数阈值以及消息队列中存储消息的最大行数等。

[0064] 在本申请的一个实施例中，可以读取消息队列元信息表，以获取多个消息队列的特征。

[0065] 在本申请的一个实施例中，可以对数据库中的表格进行通用的定义，可以将目标消息转换成字节格式，将字节格式的目标消息存储至消息表中，以使消息队列具有通用性。

[0066] 在本申请的一个实施例中，可以将目标消息序列化成二进制格式。

[0067] 在本申请的一个实施例中，数据库可以关系型数据库，如Mysql数据库等；数据库也可以是非关系型数据库(Nosql)，如Redis数据库等，数据库具有持久存储、事务、索引、高并发、高可用、运维等，使消息队列的生成和使用都非常高效。

[0068] 继续参照图3，在步骤S320中，响应于消息消费者对消息表中的目标消息的消费指令，将目标消息发送至消息消费者，并将目标消息从消息表转移至与消息表对应的确认表中。

[0069] 在本申请的一个实施例中，数据库中每个消息队列可以包含相互对应的一个消息表和一个确认表(ack_table)，以实现消息的存储和消费。

[0070] 在本申请的一个实施例中，可以基于消息表包含的消息的消息特征确定消息表特征，消息表特征可包括消息表中存储的消息数量、消息在消息表中的排序、消息表中消息的存储时长的最大值和最小值等，本实施例对此不作限定。

[0071] 在本申请的一个实施例中，可以基于确认表包含的消息的消息特征确定确认表特征，确认表特征可以包括：消息在确认表中的排序、确认表中存储的消息数量、确认表中消息的存储时长的最大值和最小值等，本实施例对此不作限定。

[0072] 在本申请的一个实施例中，可以监听消息表的消息表特征，若消息表特征发生变化且变化后的消息表特征与消息表所属的消息队列的特征不对应，则对消息消费者进行告警。

[0073] 在本申请的一个实施例中，可以监听消息表中存储的消息数量，若消息表中存储的消息数量发生变化，且变化后的消息表中存储的消息数量达到消息队列中存储消息的最大行数，则对消息消费者进行告警。

[0074] 在本申请的一个实施例中，可以监听目标消息在消息表中的存储时长，若目标消息在消息表中的存储时长发生变化，且变化后的目标消息在消息表中的存储时长达到消息队列中存储消息的时长，则对消息消费者进行告警。

[0075] 在本申请的一个实施例中，可以监听消息表中目标消息转存至确认表的转存次数，若转存次数发生变化，且变化后的转存次数达到消息队列中消息最多被转存次数，则对消息消费者进行告警。

[0076] 在本申请的一个实施例中，可以监听确认表的确认表特征，若确认表特征发生变化，且变化后的确认表特征与确认表所属的消息队列的特征不对应，则对消息消费者进行告警。

[0077] 在本申请的一个实施例中，可以监听确认表中存储的消息数量，若确认表中存储

的消息数量发生变化，且确认表中存储的消息数量达到消息队列中存储消息的最大行数，则对消息消费者进行告警。

[0078] 在本申请的一个实施例中，可以监听目标消息在确认表中的存储时长，若目标消息在确认表中的存储时长达到消息队列中存储消息的时长，则对消息消费者进行告警。

[0079] 在本申请的一个实施例中，可以监听确认表中目标消息被发送至消息消费者的发送次数，若发送次数发生变化，且变化后的发送次数达到发送次数阈值，则对消息消费者进行告警。

[0080] 在本申请的一个实施例中，可以统计目标消息在确认表中的时长，若时长达到超时阈值，则统计目标消息的发送次数，若发送次数达到发送次数阈值，则将目标消息由确认表转存至过期消息存储表(dead_message_table)中；若发送次数未达到发送次数阈值，则将目标消息由确认表转存至消息表中，以再次响应目标消息的消费指令后重新发送目标消息。

[0081] 在本申请的一个实施例中，可以获取过期消息存储表中的目标消息，以多目标消息进行修复，修复后转存至消息表中，以再次响应目标消息的消费指令后重新发送目标消息。

[0082] 在本申请的一个实施例中，若检测到过期消息存储表中的目标消息不能修复成功，可以将过期消息存储表中的目标消息删除。

[0083] 在本申请的一个实施例中，消费指令可以是排他指令，在响应消息消费者针对于目标消息的消费指令后，在消息消费者消费目标消息的过程中，禁止将目标消息发送至其他消费者。

[0084] 继续参照图3，在步骤S330中，若接收到消息消费者发送的针对目标消息的确认指令，则将消息从确认表中删除。

[0085] 在本申请的一个实施例中，可以由队列管理者(Message Queue Manager，MQ Manager)管理消息队列，一个队列管理者可以管理一个或多个消息队列，队列管理者可以进行目标消息的转存和删除处理，队列管理者可以监控数据库中消息队列的运行、管理消息的派发、维护消息队列的运行。

[0086] 在图3的实施例中，通过将消息生产者生产的目标消息存储至数据库的消息表中，响应于消息消费者对消息表中的目标消息的消费指令，将目标消息发送至消息消费者，并将目标消息从消息表转移至与消息表对应的确认表中，若接收到消息消费者发送的针对目标消息的确认指令，则将消息从确认表中删除，实现使用数据库存储消息生成消息队列，并且还通过数据库实现了向消息队列中生产消息和从消息队列中消费消息的功能，相比于使用多台服务器存储消息，能够实现使用数据库来提供消息队列功能，能够降低用户运维和使用消息队列的难度，简单易用。此外，消息生产者和消息消费者可以直接操作数据库来生产或消费消息，简便高效。

[0087] 图4示意性示出了根据本申请的一个实施例的消息处理系统的结构示意图，参照图4所示，该消息处理系统可以包括消息队列库(Message Queue Library，MQ Library)401、数据库402和队列管理者(Message Queue Manager，MQ Manager)403。消息队列库401中存储有消息生产者和消息消费者，为消息生产者提供生产接口(Produce)，为消息消费者提供消费接口(Consume)和确认接口(Confirm)；数据库402用于与消息队列库401进行消息

交互,存储有相互对应的消息表和确认表,消息表用于存储消息生产者通过生产接口发送至数据库402且未被发送至消息消费者的消息,确认表用于存储数据库402通过消费接口发送至消息消费者且未被消息消费者确认的消息,确认表接收消息消费者通过确认接口发送的确认指令后将确认指令对应的消息删除;队列管理者403用于监控数据库402及管理数据库402。

[0088] 如图4所示,在本申请的一个实施例中,消息队列库401主要包括四个指令:

[0089] (1)消息队列获取指令List<QueueName>ListQueues():获取所有可用的队列信息,用户指定队列进行Produce/Consume。实现上,打印queue_meta_table内容,也就是打印所有队列的元信息;

[0090] (2)生产指令Produce(queue_nam,message):向指定队列生产一条新消息。实现上,向队列名为queue_name的队列对应message_table末尾插入新行;

[0091] (3)消费指令List<Message>Consume(queue_name,batch_size):从指定队列消费新消息。实现上,从队列名为queue_name的队列对应message_table的头部读取一些消息,从message_table删除这些消息并移到ack_table中。这三步通过MySQL提供的Transaction实现过程的原子性。消息的消费过程是互斥的,也就是说一个队列的多个消费者不可能同时消费同一条消息,这是通过MySQL的带锁读“SELECT FOR UPDATE”实现。一条消息Message通过queue_name,message_id,version三个字段来唯一标识;

[0092] (4)确认指令Confirm():确认该消息消费成功,消息被删除。实现上,从消息对应的ack_table中删除该消息,即从queue_name对应的队列ack_table中删除message_id,version标识的消息。

[0093] 在该实施例中,消息生产客户端201和消息消费客户端202只需要通过生产接口、消费接口和确认接口即可与消息队列进行交互,而不是与存储服务器(Kfaka的broker)交互,不需要维护多个服务器来存储消息,操作上简单高效,

[0094] 继续参照图4,在本申请的一个实施例中,数据库402中可以存储有四种类型的表格:包括一个消息队列元信息表(queue_meta_table)、一个过期消息存储表(dead_message_table)、多个消息表(dead_message_table)和(message_table)对应的消息表的多个确认表(ack_table)。

[0095] 在本申请的一个实施例中,消息队列元信息表用于存放:队列名字、工作开关、最多被消费次数、Confirm超时阈值等。对该表的操作有:Manager创建/删除/修改队列属性。

[0096] 在本申请的一个实施例中,消息队列元信息表的表格定义中可以定义消息队列名称、消息队列的工作开关、消息队列的超时阈值、发送次数阈值和转存次数阈值等,具体的,消息队列元信息表的表格定义可以为:

[0097] “CREATE TABLE queue_meta_table(

[0098] queue_name VARCHAR(64)PRIMARY KEY,

[0099] enable_produce TINYINT UNSIGNED DEFAULT 1,

[0100] max_consume_count TINYINT UNSIGNED DEFAULT 3,

[0101] confirm_timeout_s INT UNSIGNED DEFAULT 600

[0102])ENGINE=InnoDB DEFAULT CHARSET=utf”。

[0103] 在本申请的一个实施例中,每个消息队列对应一个message_table和一个ack_

table。message_table存放消息队列新生产消息的数据库表格，也是消息消费过程的主要消息来源。该表主要操作有：消息生产者调用Proudce接口，向队列生产新消息，存放在message_table中，消息消费者读取message_table的消息，把消息从message_table中删除，移到ack_table。

[0104] 在本申请的一个实施例中，消息表(message_table)的表格定义中可以定义与该消息队列对应的消息特征、统计消息表中消息的消费次数、统计消息表中的消息是否被确认、获取消息生产者标识、消息的生产时间、消息从生产至被消费的时间、将消息进行二进制序列化等，具体的，message_table表格定义可以为：

[0105] “CREATE TABLE queue_0(

[0106] message_id BIGINT UNSIGNED

[0107] AUTO_INCREMENT PRIMARY KEY COMMENT’the unique id of each message in a message_queue’,

[0108] version INT UNSIGNED DEFAULT 0COMMENT’the times that the message has been consumed’,

[0109] trace_id BIGINT UNSIGNED NOT NULL COMMENT’user-defined debug info of each message’,

[0110] producer_id BIGINT UNSIGNED NOT NULL COMMENT’producer_id info of each message’,

[0111] create_time DATETIME NOT NULL COMMENT’create time when the message was inserted into message_queue’,

[0112] time_to_live DATETIME NOT NULL COMMENT’lifetime of each message’,

[0113] message LONGBLOB NOT NULL COMMENT’user data serialized in binary format’

[0114])ENGINE＝InnoDB DEFAULT CHARSET＝utf8;”。

[0115] 在本申请的一个实施例中，确认表(ack_table)用于存放等待确认(Confirm)的消息。对ack_table的主要操作有：消息Consume过程会把消息从message_table移到ack_table。消息Confirm过程会把消息从ack_table中删除。Manager定期扫描所有的ack_table，发现有消息超时未Confirm时，派发次数小于阈值(最大消息消费次数-1)，会被Manager将消息移到对应的message_table，实现消息的重新派发，当派发次数超过阈值后，会被Manager将消息移到dead_message_table。

[0116] 在本申请的一个实施例中，确认表(ack_table)的表格定义中可以定义与该确认表所在的消息队列对应的消息特征、获取消息生产者标识、消息消费者标识、交易标识、消息生产时间、消息在确认表中的时间、超时阈值、将消息进行二进制序列化等，具体的，ack_table的表格定义可以为：

[0117] “CREATE TABLE queue_0_ack(

[0118] message_id BIGINT UNSIGNED PRIMARY KEY,

[0119] version INT UNSIGNED DEFAULT 0,

[0120] trace_id BIGINT UNSIGNED NOT NULL,

[0121] consumer_id BIGINT UNSIGNED NOT NULL COMMENT’consumer_id info of this

waiting-ack message',

[0122] producer_id BIGINT UNSIGNED NOT NULL,

[0123] create_time DATETIME NOT NULL,

[0124] time_to_live DATETIME NOT NULL,

[0125] expiration_time DATETIME NOT NULL COMMENT'waiting-ack timeout of each message',

[0126] message LONGBLOB NOT NULL,

[0127] INDEX(consumer_id),

[0128] INDEX(expiration_time)

[0129])ENGINE＝InnoDB DEFAULT CHARSET＝utf8;”。

[0130] 在本申请的一个实施例中,当一条消息由于过期未Confirm,会被Manager重新派发。当派发次数超过阈值后,被认为是过期信息(dead message),存放至过期消息存储表(dead_message_table)。dead_message_table表专门存放dead_message。对该表的操作有:Manager将各个ack_table的dead_message移到该表。人工确认dead_message产生原因,根据情况对dead_message_table中的消息进行处理。

[0131] 在本申请的一个实施例中,过期消息存储表(dead_message_table)的表格定义中可以定义其管理的消息标识、消息队列标识、交易标识、消息生产者标识、消息消费者标识、消息生产时间、消息在各个表中的时间以及将消息进行二进制序列化等,具体的,dead_message_table的表格定义可以为:

[0132] “CREATE TABLE dead_message_table(

[0133] message_id BIGINT UNSIGNED AUTO_INCREMENT PRIMARY KEY,

[0134] queue_name VARCHAR(64)NOT NULL,

[0135] version INT UNSIGNED NOT NULL,

[0136] trace_id BIGINT UNSIGNED NOT NULL,

[0137] producer_id BIGINT UNSIGNED NOT NULL,

[0138] create_time DATETIME NOT NULL,

[0139] time_to_live DATETIME NOT NULL,

[0140] message LONGBLOB NOT NULL

[0141])ENGINE＝InnoDB DEFAULT CHARSET＝utf8;”。

[0142] 在该实施例中,数据库402中的表格作为消息的存储引擎具有通用的表格定义,将用户消息序列化成二进制,使数据库402中的表格不失通用性。

[0143] 继续参照图4,在本申请的一个实施例中,Manager可以主要负责队列管理功能、队列运行监控和队列过期消息的重新派发。

[0144] 在本申请的一个实施例中,队列管理功能可以包括输出队列信息、创建队列、删除队列、清空队列和修改队列属性,具体地:输出队列信息是输出queue_meta_table的所有队列信息;创建队列工作是创建message_table,ack_table,并插入该队列元信息到queue_meta_table中;删除队列与“创建队列”相反;清空队列是删除队列对应的message_table,ack_table的所有消息;修改队列属性是修改queue_meta_table队列的运行开关、最多被消费次数和Confirm超时阈值等。

[0145] 在本申请的一个实施例中，队列运行监控可以包括定期扫描所有队列的message_table和ack_table，统计其行数、最新生产时间和最早生产时间；根据用户配置，当某些信息超过阈值后，告警给接收人。

[0146] 在本申请的一个实施例中，队列过期消息的重新派发可以包括定期扫描所有队列的ack_table，当有消息超过阈值没有Confirm后，若消息派发次数小于阈值(最大消息消费次数-1)：Manager将消息移到对应的message_table，实现消息的重新派发；若消息派发次数超过阈值后：Manager将消息移到dead_message_table。

[0147] 在该实施例中，一个消息队列管理者403节点，提供队列运行监控、过期消息重新派发以及队列运维功能。该节点非常轻量级，运维简单。在图4的实施例中，通过“消息队列库401+数据库402+队列管理者403”实现了简单、高效、轻量化的消息队列系统，使消息队列的生成和维护都十分简单，并且具有通用性，可轻松复用到各个应用场景中。

[0148] 以下介绍本申请的装置实施例，可以用于执行本申请上述实施例中的消息处理方法。对于本申请装置实施例中未披露的细节，请参照本申请上述的消息处理方法的实施例。

[0149] 图5示意性示出了根据本申请的一个实施例的消息处理装置的框图。

[0150] 参照图5所示，根据本申请的一个实施例的消息处理装置500，包括存储模块501、转移模块502和删除模块503。

[0151] 在本申请的一些实施例中，基于前述方案，存储模块501配置为将消息生产者生产的目标消息存储至数据库的消息表中；转移模块502配置为响应于消息消费者对消息表中的目标消息的消费指令，将目标消息发送至消息消费者，并将目标消息从消息表转移至与消息表对应的确认表中；删除模块503配置为若接收到消息消费者发送的针对目标消息的确认指令，则将消息从确认表中删除。

[0152] 在本申请的一个实施例中，基于前述方案，存储模块501配置为：获取目标消息的消息特征；基于消息特征，将目标消息存储至与消息特征对应的消息表。

[0153] 在本申请的一个实施例中，基于前述方案，数据库中包含有多个消息队列，每个消息队列包含相互对应的一个消息表和一个确认表，存储模块501配置为：从数据库的消息队列元信息表中，获取多个消息队列的特征；基于消息特征和多个消息队列的特征，在多个消息队列中选取与目标消息的消息特征对应的消息队列；将目标消息存储至与目标消息的消息特征对应的消息队列包含的消息表中。

[0154] 在本申请的一个实施例中，基于前述方案，转移模块502还配置为：在若接收到消息消费者返回的针对目标消息的确认指令，则将消息从确认表中删除之前，监听消息表的消息表特征，若消息表特征发生变化且变化后的消息表特征与消息表所属的消息队列的特征不对应，则对消息消费者进行告警；监听确认表的确认表特征，若确认表特征发生变化且变化后的确认表特征与确认表所属的消息队列的特征不对应，则对消息消费者进行告警。

[0155] 在本申请的一个实施例中，基于前述方案，转移模块502还配置为：在若接收到消息消费者返回的针对目标消息的确认指令，则将消息从确认表中删除之前，统计目标消息在确认表中的时长；若时长达到超时阈值，则统计目标消息被发送至所述消息消费者的发送次数；若发送次数达到发送次数阈值，则将目标消息由确认表转存至过期消息存储表中。

[0156] 在本申请的一个实施例中，基于前述方案，转移模块502还配置为：若发送次数未达到发送次数阈值，则将目标消息由确认表转存至消息表中。

[0157] 在本申请的一个实施例中，基于前述方案，存储模块501配置为：将目标消息转换成字节格式；将字节格式的目标消息存储至消息表中。

[0158] 在本申请的一个实施例中，基于前述方案，转移模块502配置为：在消息消费者消费目标消息的过程中，禁止将目标消息发送至其他消费者。

[0159] 所属技术领域的技术人员能够理解，本申请的各个方面可以实现为系统、方法或程序产品。因此，本申请的各个方面可以具体实现为以下形式，即：完全的硬件实施方式、完全的软件实施方式(包括固件、微代码等)，或硬件和软件方面结合的实施方式，这里可以统称为“电路”、“模块”或“系统”。

[0160] 下面参照图6来描述根据本申请的这种实施方式的电子设备60。图6显示的电子设备60仅仅是一个示例，不应对本申请实施例的功能和使用范围带来任何限制。

[0161] 如图6所示，电子设备60以通用计算设备的形式表现。电子设备60的组件可以包括但不限于：上述至少一个处理单元61、上述至少一个存储单元62、连接不同系统组件(包括存储单元62和处理单元61)的总线63、显示单元64。

[0162] 其中，所述存储单元存储有程序代码，所述程序代码可以被所述处理单元61执行，使得所述处理单元61执行本说明书上述“实施例方法”部分中描述的根据本申请各种示例性实施方式的步骤。

[0163] 存储单元62可以包括易失性存储单元形式的可读介质，例如随机存取存储单元(RAM)621和/或高速缓存存储单元622，还可以进一步包括只读存储单元(ROM)623。

[0164] 存储单元62还可以包括具有一组(至少一个)程序模块625的程序/实用工具624，这样的程序模块625包括但不限于：操作系统、一个或者多个应用程序、其它程序模块以及程序数据，这些示例中的每一个或某种组合中可能包括网络环境的实现。

[0165] 总线63可以为表示几类总线结构中的一种或多种，包括存储单元总线或者存储单元控制器、外围总线、图形加速端口、处理单元或者使用多种总线结构中的任意总线结构的局域总线。

[0166] 电子设备60也可以与一个或多个外部设备(例如键盘、指向设备、蓝牙设备等)通信，还可与一个或者多个使得用户能与该电子设备60交互的设备通信，和/或与使得该电子设备60能与一个或多个其它计算设备进行通信的任何设备(例如路由器、调制解调器等等)通信。这种通信可以通过输入/输出(I/O)接口65进行。并且，电子设备60还可以通过网络适配器66与一个或者多个网络(例如局域网(LAN)，广域网(WAN)和/或公共网络，例如因特网)通信。如图所示，网络适配器66通过总线63与电子设备60的其它模块通信。应当明白，尽管图中未示出，可以结合电子设备60使用其它硬件和/或软件模块，包括但不限于：微代码、设备驱动器、冗余处理单元、外部磁盘驱动阵列、RAID系统、磁带驱动器以及数据备份存储系统等。

[0167] 通过以上的实施方式的描述，本领域的技术人员易于理解，这里描述的示例实施方式可以通过软件实现，也可以通过软件结合必要的硬件的方式来实现。因此，根据本申请实施方式的技术方案可以以软件产品的形式体现出来，该软件产品可以存储在一个非易失性存储介质(可以是CD-ROM，U盘，移动硬盘等)中或网络上，包括若干指令以使得一台计算设备(可以是个人计算机、服务器、终端装置、或者网络设备等)执行根据本申请实施方式的方法。

[0168] 根据本申请一个实施例，还提供了一种计算机可读存储介质，其上存储有能够实现本说明书上述方法的程序产品。在一些可能的实施方式中，本申请的各个方面还可以实现为一种程序产品的形式，其包括程序代码，当所述程序产品在终端设备上运行时，所述程序代码用于使所述终端设备执行本说明书上述“示例性方法”部分中描述的根据本申请各种示例性实施方式的步骤。

[0169] 根据本申请一个实施例，用于实现上述方法的程序产品可以采用便携式紧凑盘只读存储器(CD-ROM)并包括程序代码，并可以在终端设备，例如个人电脑上运行。然而，本申请的程序产品不限于此，在本文件中，可读存储介质可以是任何包含或存储程序的有形介质，该程序可以被指令执行系统、装置或者器件使用或者与其结合使用。

[0170] 所述程序产品可以采用一个或多个可读介质的任意组合。可读介质可以是可读信号介质或者可读存储介质。可读存储介质例如可以为但不限于电、磁、光、电磁、红外线、或半导体的系统、装置或器件，或者任意以上的组合。可读存储介质的更具体的例子(非穷举的列表)包括：具有一个或多个导线的电连接、便携式盘、硬盘、随机存取存储器(RAM)、只读存储器(ROM)、可擦式可编程只读存储器(EPROM或闪存)、光纤、便携式紧凑盘只读存储器(CD-ROM)、光存储器件、磁存储器件、或者上述的任意合适的组合。

[0171] 计算机可读信号介质可以包括在基带中或者作为载波一部分传播的数据信号，其中承载了可读程序代码。这种传播的数据信号可以采用多种形式，包括但不限于电磁信号、光信号或上述的任意合适的组合。可读信号介质还可以是可读存储介质以外的任何可读介质，该可读介质可以返回、传播或者传输用于由指令执行系统、装置或者器件使用或者与其结合使用的程序。

[0172] 可读介质上包含的程序代码可以用任何适当的介质传输，包括但不限于无线、有线、光缆、RF等等，或者上述的任意合适的组合。

[0173] 可以以一种或多种程序设计语言的任意组合来编写用于执行本申请操作的程序代码，所述程序设计语言包括面向对象的程序设计语言—诸如Java、C++等，还包括常规的过程式程序设计语言—诸如“C”语言或类似的程序设计语言。程序代码可以完全地在用户计算设备上执行、部分地在用户设备上执行、作为一个独立的软件包执行、部分在用户计算设备上部分在远程计算设备上执行、或者完全在远程计算设备或服务器上执行。在涉及远程计算设备的情形中，远程计算设备可以通过任意种类的网络，包括局域网(LAN)或广域网(WAN)，连接到用户计算设备，或者，可以连接到外部计算设备(例如利用因特网服务提供商来通过因特网连接)。

[0174] 此外，上述附图仅是根据本申请示例性实施例的方法所包括的处理的示意性说明，而不是限制目的。易于理解，上述附图所示的处理并不表明或限制这些处理的时间顺序。另外，也易于理解，这些处理可以是例如在多个模块中同步或异步执行的。

[0175] 应当理解的是，本申请并不局限于上面已经描述并在附图中示出的精确结构，并且可以在不脱离其范围执行各种修改和改变。本申请的范围仅由所附的权利要求来限制。

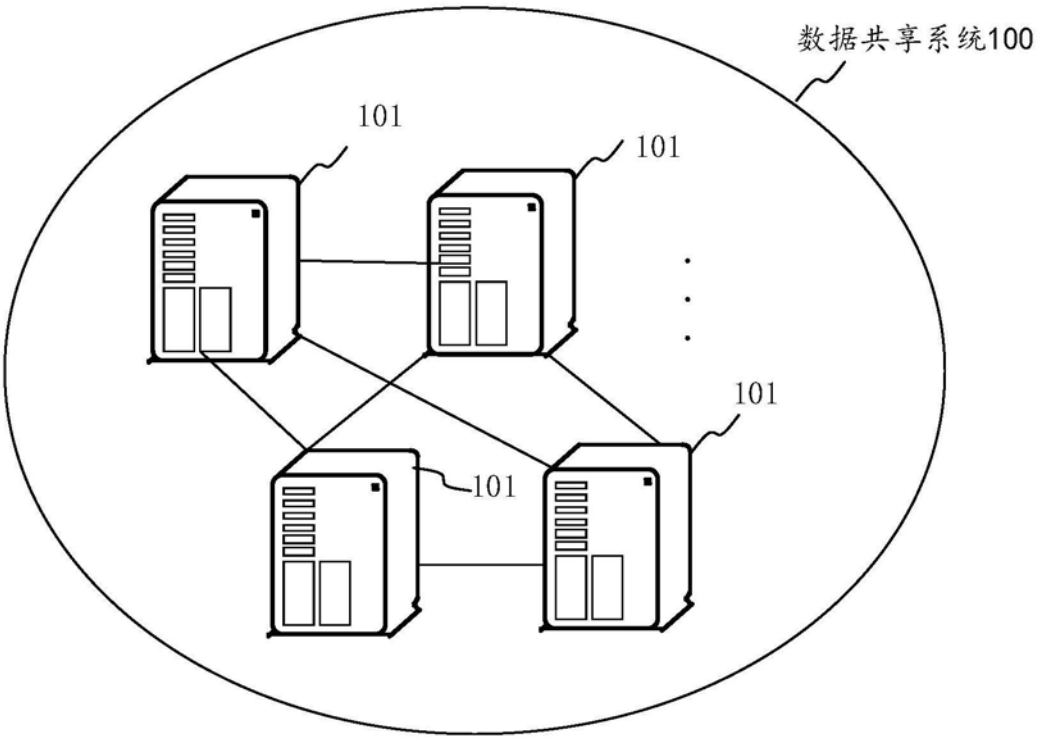

数据共享系统100
101
101
101
101

图1A

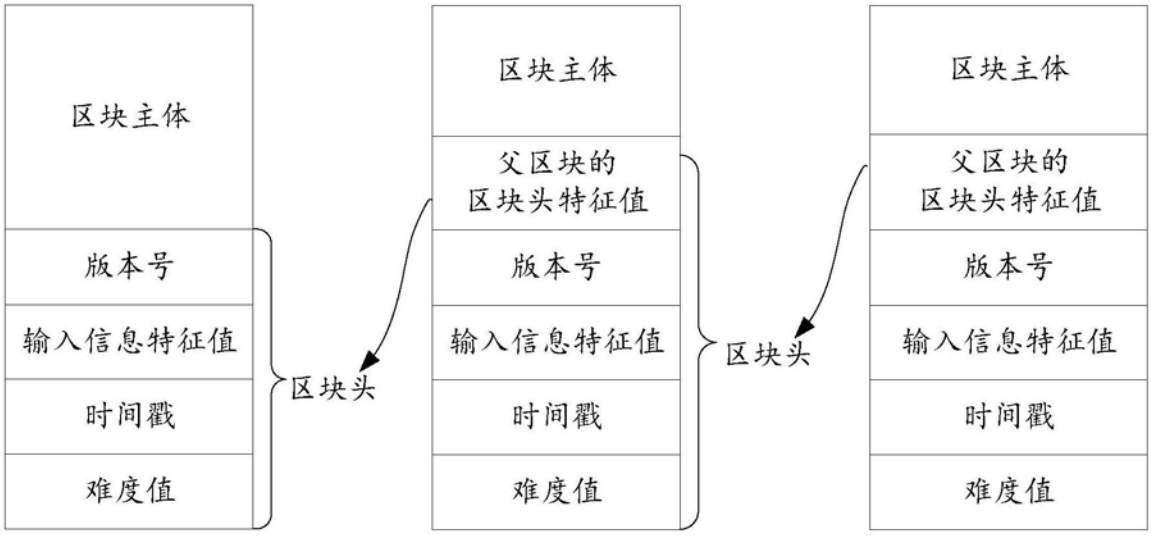

区块主体
版本号
输入信息特征值
时间戳
难度值
区块头
区块主体
父区块的
区块头特征值
版本号
输入信息特征值
时间戳
难度值
区块头
区块主体
父区块的
区块头特征值
版本号
输入信息特征值
时间戳
难度值

图1B

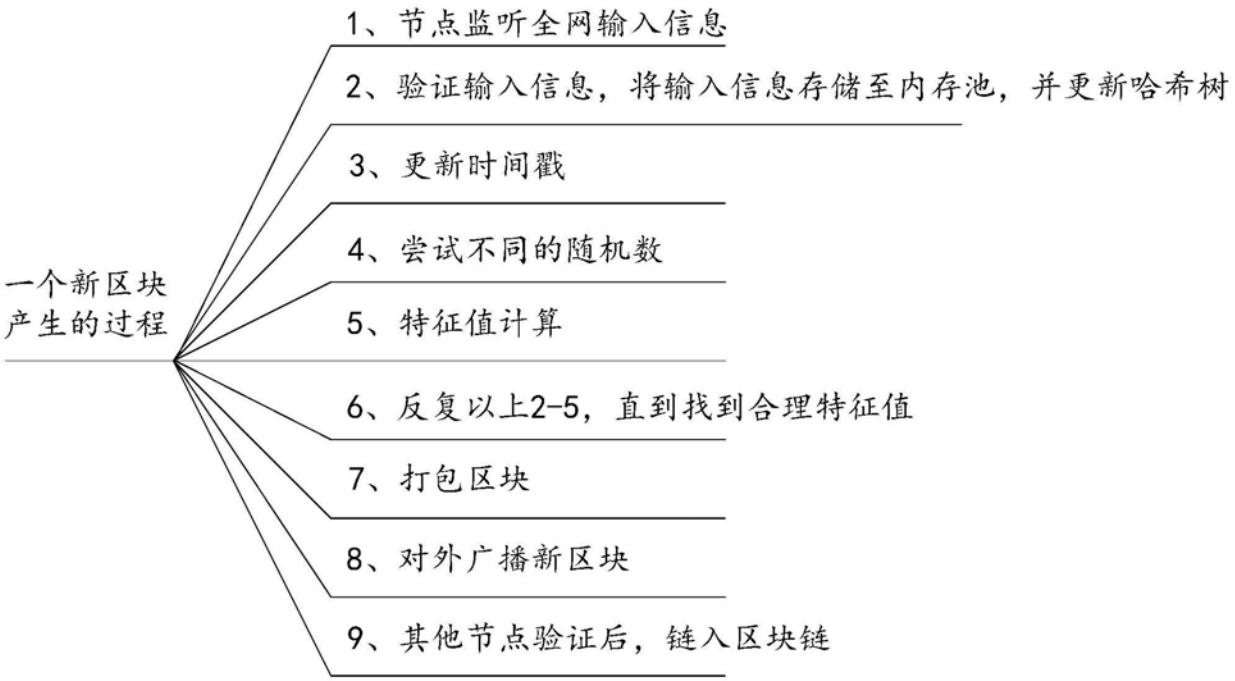

一个新区块
产生的过程
1、节点监听全网输入信息
2、验证输入信息，将输入信息存储至内存池，并更新哈希树
3、更新时间戳
4、尝试不同的随机数
5、特征值计算
6、反复以上2-5，直到找到合理特征值
7、打包区块
8、对外广播新区块
9、其他节点验证后，链入区块链

图1C

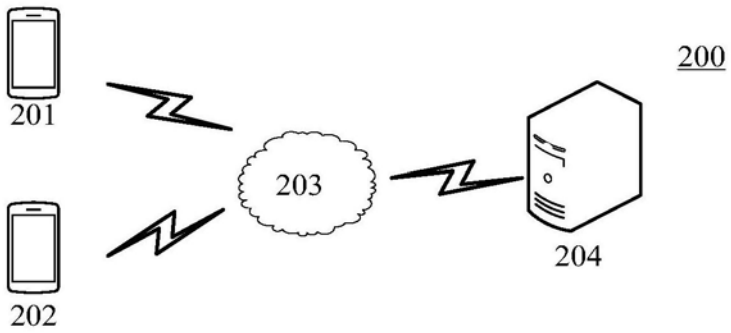

200
201
202
203
204

图2

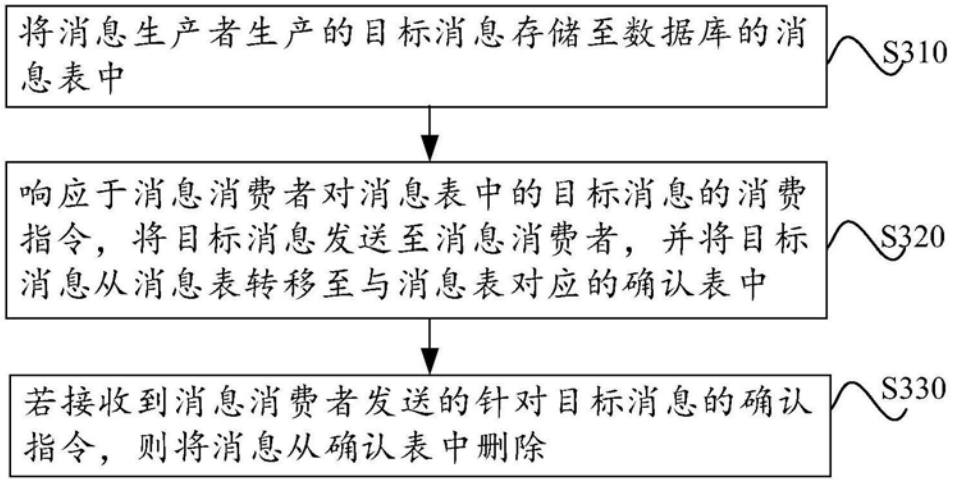

将消息生产者生产的目标消息存储至数据库的消息表中
S310
响应于消息消费者对消息表中的目标消息的消费指令，将目标消息发送至消息消费者，并将目标消息从消息表转移至与消息表对应的确认表中
S320
若接收到消息消费者发送的针对目标消息的确认指令，则将消息从确认表中删除
S330

图3

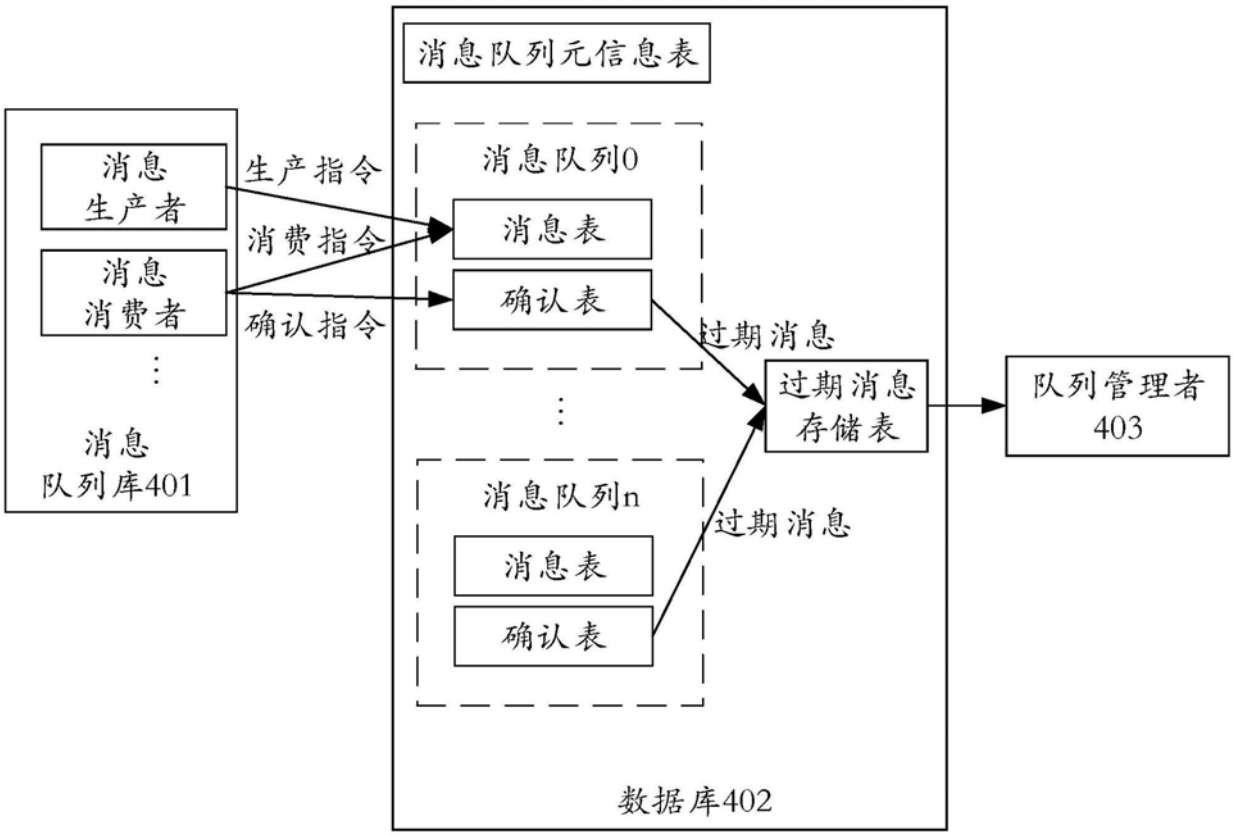

消息队列元信息表
消息
生产者
消息
消费者
消息
队列库401
生产指令
消费指令
确认指令
消息队列0
消息表
确认表
过期消息
过期消息
存储表
队列管理者
403
消息队列n
消息表
确认表
过期消息
数据库402

图4

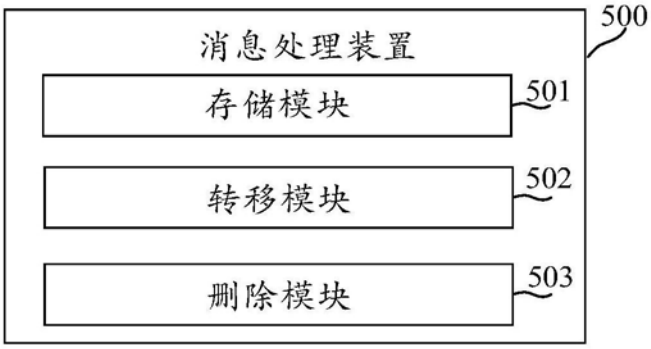

消息处理装置
500
存储模块
501
转移模块
502
删除模块
503

图5

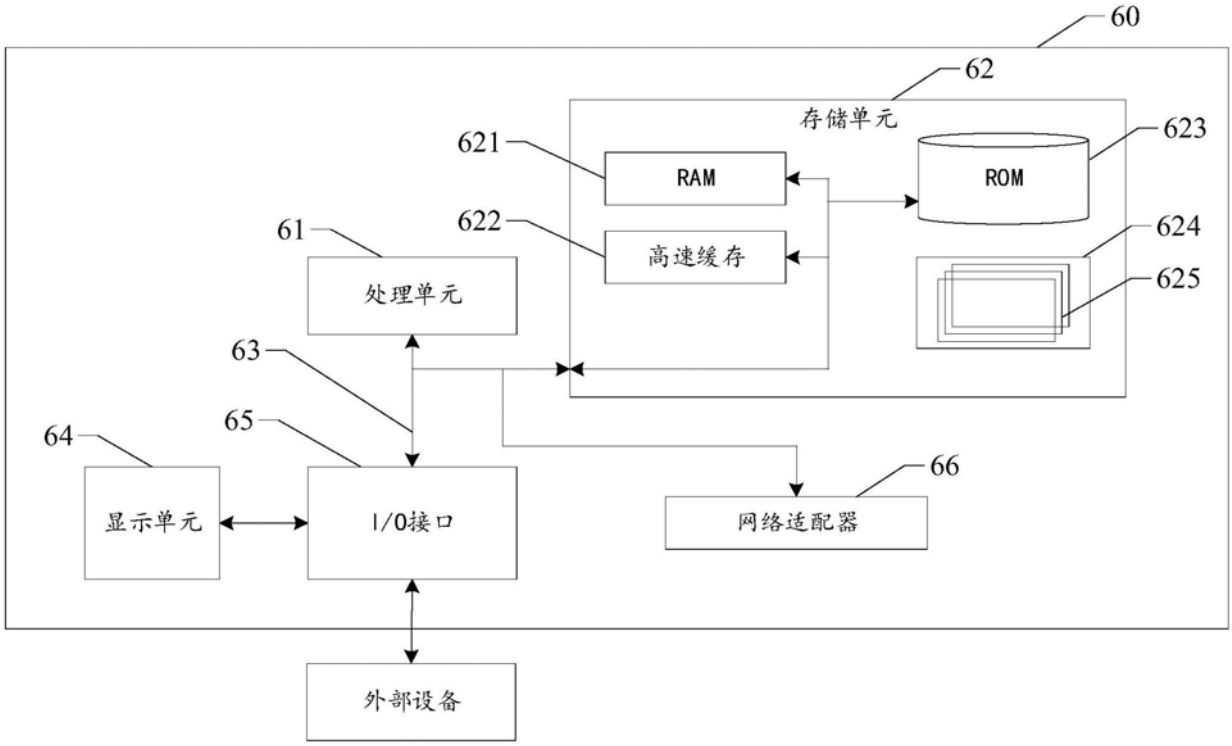

60
62
存储单元
621
RAM
623
ROM
622
高速缓存
624
625
61
处理单元
63
65
64
显示单元
I/O接口
66
网络适配器
外部设备

图6